

**Teoría de las Comunicaciones**  
28 de Noviembre de 2018  
2do Parcial



Departamento de Computación  
Facultad de Ciencias Exactas y Naturales  
Universidad de Buenos Aires

Apellido: <input type="text"/>	Orden: X	Hojas > 4	Ej.1 1	Ej.2 1	Ej.3 1	Ej.4 1	
Nombres: <input type="text"/>	LU: <input type="text"/>	Calif. >	B	B	B	B	Final: A+

Todas las respuestas se consideran válidas solo si están debidamente justificadas.  
Entregar cada ejercicio en hojas separadas.

### Ejercicio 1

Dada la siguiente captura de segmentos TCP que empieza con ambos extremos en ESTABLISHED:

No.	Source	Destination	Info
1	192.168.100.35	192.168.100.40	8443 > 33242 [ACK] Seq=50 Ack=110 Len=0
2	192.168.100.40	192.168.100.35	33242 > 8443 [PSH,ACK] Seq=110 Ack=50 Len=80
3	192.168.100.35	192.168.100.40	8443 > 33242 [ACK] Seq=50 Ack=190 Len=10
4	192.168.100.40	192.168.100.35	33242 > 8443 [FIN,ACK] Seq=190 Ack=50 Len=0
5	192.168.100.35	192.168.100.40	8443 > 33242 [FIN,ACK] Seq=60 Ack=190 Len=0
6	192.168.100.35	192.168.100.40	8443 > 33242 [ACK] Seq=61 Ack=191 Len=0
7	192.168.100.40	192.168.100.35	33242 > 8443 [ACK] Seq=191 Ack=61 Len=0

- Indicar la secuencia de estados TCP atravesados por cada socket suponiendo que no se pierde ningún segmento.
- Ahora suponga que se pierde el segmento 7, explique los eventos que suceden y los cambios de estados en ambos extremos, hasta que llegan al estado CLOSED.

### Ejercicio 2

En un momento dado, Cosme descarga su casilla de correo en su PC desde su *user agent* y visualiza el siguiente mail.

```
To: cosme@fulanito.com.ar
From: "ofertas@vacaciones.com.ar" <ofertas@vacaciones.com.ar>
Reply-to: "no-reply@vacaciones.com.ar" <no-reply@vacaciones.com.ar>
Subject: Muchos viajes muy baratos!
MIME-Version: 1.0
Content-Type: text/html; charset = "iso-8859-1"
```

```
<html> <head></head>
<body>
  Felices vacaciones!!!
  <br />
  <br />
  <a href="http://ads.vacaciones.com.ar/comprar.php">Viaja por el mundo!!!</a><br />
  <br />
  <br />
</div>
</body>
</html>
```

- Si el *user agent* usa POP3 y HTTP/1.1 para ver el mail. ¿Cuántas conexiones TCP se abrieron para descargar y visualizar el mail?
- ¿Cuántas consultas DNS y de qué tipo desencadena la visualización del mail? Asumir que todas las caches están vacías y que hay un Servidor Autoritativo por zona.

### Ejercicio 3

Una conexión recién establecida tiene un  $RTT=100ms$  y debe transmitir 50KB. Al principio, el receptor anuncia una *Advertised Window* de 64KB y se sabe que el proveedor de servicio del host emisor limita la velocidad descartando todos los segmentos de una ráfaga si se envían 32KB o más por RTT.

- ¿Cuántos datos lleva transmitidos con éxito (i.e.: datos que ya no están "en vuelo") a los 550ms?
- Si a partir de los 700ms de iniciada la transferencia, los ACKs que arriban al emisor tienen una *Advertised Window* de 16KB ¿Cuanto vale la CWND una vez finalizada la transferencia?

### Ejercicio 4

Una organización nos pide que implementemos una política de seguridad para su red que tiene las siguientes características:

- 25 hosts de los empleados con información crucial de la compañía.
  - 1 webserver accesible desde Internet que usa HTTPS solamente.
  - 1 servidor DNS del dominio de la compañía y que además se encarga de resolver todas las operaciones de DNS de la red.
  - 1 servidor Proxy para que los usuarios puedan acceder a la web.
- Presente un esquema gráfico de la red y detalle todas las reglas de firewall necesarias para implementar la política de seguridad especificada.
  - La compañía desea que sólo usuarios autenticados puedan acceder a recursos Web en Internet conectandose mediante HTTPS con el Proxy. Explique cómo hacer para garantizar la autenticidad de los usuarios del lado del Proxy, aclarando dónde se instalarían los certificados digitales.

1) a) En base a la secuencia, en la que ambos extremos inician en ESTABLISHED, notamos que en los primeros 3 segmentos correspondientes a transferencia de datos los estados de ellos no varían. ✓

Luego, mirando al 4 notamos que el host A (lo tomaremos como 192.168.100.40:3342) le envió un mensaje con los flags FIN+ACK a B

(el extremo 192.168.100.35:8493). Esto hace que A pase de ESTABLISHED a FIN\_WAIT\_1, puesto que el ACK está para reconocer los datos enviados. ✓

en el segmento 3. Por otra parte, B recibe el segmento y se prepara para hacer la transición.

Seguidamente, B envía un segmento con los flags FIN+ACK, de manera de pasar directo al estado LAST\_ACK. A recibe este segmento y en respuesta envía.

Ahora, si nos fijamos en el segmento 5 vemos que B le envía a A un mensaje con los flags FIN+ACK para por su número de ACK notamos que

no se envía en respuesta al segmento anterior, puesto que el flag FIN consume el número de secuencia. Esto nos dice que en realidad se está produciendo

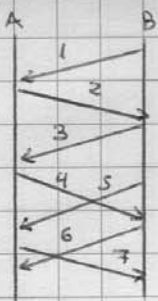
un cierre simultáneo, y B pasa a FIN\_WAIT\_1. ✓

Tras recibir el segmento de B, A envía el mensaje 6 con un flag ACK, indicando que reconoce el pedido, y pasa a CLOSING. Por otra parte, B recibe

el flag FIN pasando al estado CLOSING al enviar el ACK del segmento 7. Cuando cada extremo recibe respectivamente el ACK de su extremo contrario, → FALTA SUSTRAYAR.

no, pasará a TIME\_WAIT donde luego de un tiempo dado cerrará la conexión pasando a CLOSED. ✓

Esquema:



b) En caso de perderse el segmento 7 vemos que A pasará al estado TIME\_WAIT pero B se mantendrá en CLOSING, es

perando el eventual ACK. Frente a ella pueden suceder dos escenarios una vez que tras un tiempo de time out B decida

retransmitir el segmento 5 con el flag FIN+ACK por si no llegó. ✓

1- El tiempo de espera de TIME\_WAIT no se agotó al recibirlo y reenvió su ACK de manera que B pueda pasar al

estado TIME\_WAIT (asumiendo que este nuevo segmento no se pierde). ✓

2- El tiempo de espera de TIME\_WAIT se agotó y A pasó a CLOSED, por lo que al recibir el mensaje se envió uno de respuesta con el flag ACK

para que finalice la conexión. ✓



2) a) Considerando que para visualizar el mail el user agent debe acceder a su casilla de correo con POP3, luego este debe establecer una conexión TCP con su servidor de correo entrante ( $\langle \text{user IP}, 1024, \text{servidor mail}, 110, \text{tcp} \rangle$ ), asumiendo que ya se realizaron las consultas DNS para resolver su dirección IP (los confiamos, loagut).

Al recibirla vemos que para visualizarlo necesita cargar los recursos de los dominios ads.vacaciones.com.ar y www.vacaciones.com.ar.

Esto hace que deba establecer una conexión TCP con los servidores web correspondientes de manera de hacer los respectivos pedidos HTTP. Ahora, como la

versión de HTTP es 1.1, luego para cada conexión no hace falta reiniciarlo para obtener cada recurso de su dominio. Esto nos añade dos flujos.

TCP más:  $\langle \text{user IP}, 1024, \text{ads web IP}, 80, \text{tcp} \rangle$  y  $\langle \text{user IP}, 1024, \text{www web IP}, 80, \text{tcp} \rangle$ .

Esto nos dice que se efectúan 3 conexiones TCP para visualizar el mail y des cargarlo.

b) Mirando ahora los consultas DNS en el pedido, vemos que la primera que surge es la que establece el user agent con su resolver por la entrada

MX del dominio de mail que maneja, de la forma: request domainmail.com MX. Esta consulta es recursiva y el resolver puede no tenerla

cacheada, por lo que realiza una consulta iterativa desde el resolver al servidor raíz para esta entrada (es la misma de antes).

Con su respuesta probablemente se recibe la dirección del servidor de .com, con lo cual

b) Para visualizar el mail una vez que lo tiene des cargado vemos que el user agent necesita realizar pedidos HTTP a los dominios de los recursos.

Para ello le envía a su resolver el pedido request ads.vacaciones.com.ar de forma recursiva. Este no posee cacheada la entrada pedida, por

lo que forwardea la consulta al servidor raíz de forma iterativa.

Este responde con la dirección IP de .ar (asumiendo que al ser de alto nivel de dominio no está en un name server) y con su respuesta realiza otra

consulta iterativa, pero esta vez a la IP recibida, donde obtiene la dirección correspondiente a .com.ar (a su name server). Seguidamente se realiza

una tercera consulta iterativa para a.com.ar por el servidor autoritativo de la zona vacaciones.com.ar. y es a este que le realiza una cuarta

consulta iterativa y obtiene la IP de ads.vacaciones.com.ar que responde al user agent.

Ahora, por los recursos en el dominio www.vacaciones.com.ar vemos que el user agent nuevamente necesita resolver la IP de su servidor web

y accede a su resolver con una consulta recursiva. Si bien este no lo posee cacheado, lo que sí tiene es la del servidor autoritativo de su zona, por

lo que realiza una consulta iterativa a este de la forma request www.vacaciones.com.ar A y de este obtiene la IP del servidor web, que luego

le responde al user agent.

Con esto, vemos que se realizaron 7 consultas, 2 de las cuales fueron recursivas del user agent a su resolver

3a) Considerando la conexión con  $RTT = 100ms$ , Advertised Window =  $RWND = 64KB$  y congestión por ráfagas de  $32KB$  a más por RTT, vemos que se quieren transmitir  $50KB$  siendo la conexión recién establecida, es decir,  $CWND = 4KB$  y  $SSTHRESH = 64KB$ . Con ella, vemos que se obtiene la siguiente tabla, que contaremos hasta el RTT 6 (por SSMs):

RTT	CWND	SSTHRESH	RWND	Last Byte Sent	Last Byte Ack	Flight Size
1	4	64	64	4	0	4
2	8	64	64	12	4	8
3	16	64	64	28	12	16
4	32	64	64	50	28	22
5	50	64	64	50	50	0

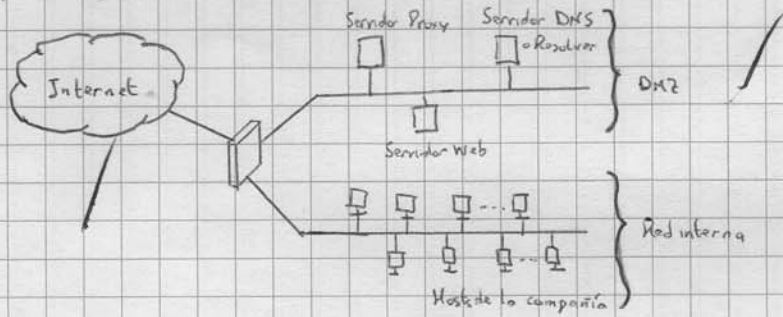
Como vemos, en el primer RTT se transmiten  $4KB$ s correspondientes a lo que entra en  $CWND = IW = 2 \times MSS = 4KB$ . Estando en SS (Slow Start), en el siguiente RTT se reconocen los paquetes enviados y  $CWND$  pasa a  $8KB$ . Análogamente en RTT 3 pasa a  $16KB$  y hasta este punto se recuperan  $12KB$ .

Ahora, en RTT 4 restan enviar  $22KB$  siendo que acumulativamente se enviaron  $28KB$  de  $50KB$ . Esto hace que más allá de que la ventana de congestión posea ser de  $32KB$ , solo se envíen  $22KB$ , por lo que el proveedor de servicio del host no los descarta y son reconocidos en el RTT siguiente. Esto nos indica que al no haber más por transmitir, ya no haya bytes en vuelo y para los  $50ms$  se hayan reconocido los  $50KB$ .

b) Considerando que la transferencia finalizó en RTT 5, es decir, a los  $500ms$ , a los  $700ms$  el valor de  $CWND$  es el mismo:  $50KB$

X SK

4 a) Considerando que en la organización se tienen hosts con información crucial, 1 web server accesible por HTTPS, 1 servidor DNS encargado de resolver las operaciones DNS de la red y de la compañía, y un servidor Proxy que le permita a los usuarios de la red acceder a Internet, vemos que podemos graficar la red de la forma:



De aquí vemos que en el firewall deseamos que de la Red interna se puedan enviar paquetes al servidor Proxy para acceder a la web (lo cual incluye al mismo web server de la compañía) y al resolver en caso de necesitar consultar por la IP de Proxy por DNS.

Desde DMZ se debe habilitar el paso de conexiones a Internet por parte del Servidor DNS para resolver consultas que no están en su dominio, del servidor Proxy para acceder a servidores web de Internet. Por otra parte, de Internet se dejan pasar las conexiones a DMZ para acceder al resolver por consultas DNS, y acceder al servidor web para conectarse a través de HTTPS. Con ello obtenemos la tabla

	Red Interna	DMZ	Internet
Red Interna		Proxy → HTTPS/HTTP Resolver → DNS	DROP
DMZ	DROP		Resolver → DNS Proxy → HTTP
Internet	DROP	Resolver → DNS Web server → HTTPS	

y con esto definimos las reglas de la forma:

- < Red Interna IP, >1024, Proxy IP, puerto (proxy), tcp > → consideramos que tanto las consultas HTTP como HTTPS pueden enviarse al mismo puerto
  - < Red Interna IP, >1024, Resolver IP, 53, udp >
  - < Resolver IP, >1024, Internet IP, 53, udp >
  - < Proxy IP, >1024, Internet IP, 80, tcp >
  - < Internet IP, >1024, Resolver IP, 53, udp >
  - < Internet IP, >1024, Web server, puerto (HTTPS), tcp >
- (Notemos como las reglas se definen asumiendo que el Firewall es stateful)

b) Para que solo los usuarios autenticados puedan acceder a Internet al conectarse via HTTPS con el Proxy, vemos que podemos garantizar la autenticidad de los usuarios por parte del lado del Proxy a través de un sistema de criptografía asimétrica, en donde para cada sesión el Proxy posee una clave pública conocida por este y el cliente, mientras que el cliente posee una clave privada. De esa manera, cuando recibe un pedido, para asegurarse de que el cliente efectivamente lo envió vemos que puede efectuar el método challenge response y enviarle al cliente un texto a encriptar con su clave privada. Este responde luego con su versión encriptada y luego Proxy lo desencripta con la clave pública. Si efectivamente coinciden, luego el usuario fue autenticado y el pedido procede. Si no se descarta. Nótese como esto implica que antes los usuarios puedan

compartir con el proxy una clave pública y no necesariamente se puede tener que el proxy tenga un certificado digital autenticado por una entidad ~~autorizada~~ autorizada para que el cliente pueda autenticar al proxy.

o con un certificado digital